



SLA Monitoring for Service Providers

White Paper

Turning SLA Monitoring into a Source of Income for Service Providers

Introduction

When a brand new service appears on the market, early adopters are usually more forgiving when checking the service availability. As time passes, new customers, who are usually more demanding, adopt the new service. These new customers expect the service to be of high quality and to have high availability immediately after purchase.

Regarding computer and network based services, many services are critical to the operation of the customer, and their lack of availability can be very costly to the customer even if it is only for a few minutes. These customers choose their supplier by the service level agreement they are willing to offer.

Service Level Agreement

A Service Level Agreement (SLA) is usually a negotiated agreement between two parties where one is the customer and the other is the service provider. This can be a formal legally binding contract or an informal "contract".

The SLA records a common understanding about the services, priorities, responsibilities, guarantees, and warranties. Each area of the service scope should have a "level of service" defined.

The SLA may specify the levels of availability, serviceability, performance, operation, or other attributes of the service, such as, billing. The "level of service" can also be specified as "target" and "minimum," to inform the customers what to expect (the minimum), while providing a measurable (average) target value that shows the level of the organization performance.

In some contracts, penalties may be agreed upon in the case of non-compliance of the SLA. It is important to note that the "agreement" relates to the services the customer receives, and not how the service provider delivers that service.

SLAs are more common as part of a contract with corporate customers. However, this practice is spreading and it is now becoming more common, even for a home customer, to engage a service



provider by including an SLA in a wide range of service contracts in practically all industries and markets. Internal departments (such as IT) in larger organization have adopted the idea of using SLAs with their "internal" customers — users in other departments within the same organization. One benefit of this is to enable the quality of service to be bench-marked across multiple locations or between different business units. This internal bench-marking can also be used to mark tests and provide a value comparison between an in-house department and an external service provider.

SLAs are, by nature, "output" based — the result of the service as received by the customer is the subject of the "agreement". There are different levels of SLAs, depending on the type of the customer, and his willingness to pay for the service. For example, Platinum customers are offered higher SLAs than Gold or Silver customers.

Proactive SLA Measurement

Measuring SLAs depends on the type of service supplied. In this paper we will focus on services that can be monitored remotely. Examples of such services are:

- Availability of a network based application:
 - A web server
 - A mail server
 - A banking service or any application accessed via the network
- Availability of a physical component:
 - A communication line
 - A router
 - A cable modem
 - A UPS device
 - An ATM

To monitor such components, we need a tool that is able to:

- Assign a component to a given service level based on its status
 - A component might have a range of statuses that might be assigned to different service levels.
 - The statuses must be ordered so that when a higher status for a component is discovered, the component is moved to the right service level.
- Query the components at a predefined frequency



- The tool should be scalable to support monitoring components of any size, at whatever frequency needed.
- Extract the needed information to determine the status of each monitored component
- Generate reports on the components needing attention based on their service level
 - There should be a query based mechanism to view any problematic components.
- Generate historical reports of statuses of given components
- Define what the SLA of a component is, and when it is bridged
 - The percentage of time a component is in a given status, before the agreement is bridged
- Allow the user to change the status of a component manually and automatically
 - This is used in order to change the status of a component to indicate that it is being handled.
- Have a historical report of the states of a given component

Monitoring Needs

An SLA monitoring tool must be a strong monitoring platform. The number of SLA monitored components is growing at an exponential rate and the components are getting more and more complicated. These components can be either hardware devices or software applications.

Increasing Number of Monitored Parameters

In the past it was enough to know if a component was up or not. Today we can not suffice with this information alone. An SLA is determined by much more detailed information on each component. For example, we want to know:

- What its performance and response time is
- What resources it uses, and how much of each resource it uses
- What its response time is
- Whats its jitter is

and so on.

Multi-Protocol Parameter Collection

More intelligent components do not lead to more standardization in accessing the required parameters. Any SLA monitoring solution should support multiple methods for collecting data.

Examples of such protocols are:

- SNMP
- ICMP
- WMI
- HTTP



- SQL
- Telnet
- SSH

and more.

Simple Installation and Operation

As the SLA monitoring operations get more and more complex, the need to simplify their installation and configuration grows. It is not very practical, operation and management wise, to have a monitoring agent for every component or small group of components, when the number of monitored components grows rapidly. The ideal solution is a centralized SLA monitoring product, which is able to perform its required actions from a single point, however, it is clear that this approach will fail on scale.

Hence, the optimal solution is a centralized SLA monitoring platform with as few agents as possible, when every agent monitors as many components as possible. This simplifies the installation and operation of this kind of platform and enables scale.

Complex Correlations and Conditions on the Monitored Parameters

The requirements for conditions that need to be tested on the monitored components have also changed dramatically in the past few years. In the past events were based on status changes and threshold passing. Today there is a need to correlate multiple parameters' behavior over time, in order to detect anomalies in the components, alert true problems, and not create any false alarms.

Flexibility

As the number of parameters (that can be monitored) increase exponentially, the number of possible correlations that make sense increase exponentially too. It is clear that shelf products cannot prepare an answer to all the possible needs of all enterprises in advance. Hence, a flexible monitoring solution is needed which easily enables adding more monitoring parameters, correlation rules, and conditions.

Scalability

Scale can be measured by this relation:

$$\begin{aligned} & \text{monitored components} \times \\ & \text{average parameters count} \times \\ & \text{average monitoring frequency} \times \\ & \text{average affective correlation rules} \end{aligned}$$

This equation only gives a rough estimate to the scale required by a monitoring solution. As can be seen from the previous equation, there are four major aspects of scale that should be taken into consideration.

Component Count Scale

As previously stated, the number of intelligent components for both hardware and software in the enterprise increases exponentially. Any monitoring solution must address this increase in a way that does not increase the monitoring cycle time.

Also, the monitored nodes should not effect each other. For example, if a given node does not respond, it should not effect the time the other node is monitored.

Collected Parameters Count Scale

Each component has multiple parameters that can be collected from it. Every monitoring solution must enable collecting the increasing number of parameters from the different components.

Monitoring Frequency

In the past it was good enough to perform certain checks every day or every hour. Today it is required to know of changes to parameters online, as they occur. A monitoring solution must support short monitoring cycles.

Correlation Rules

Today it is not enough to just collect data. Checks must be performed on the collected data, and some of these checks can be quite complicated. A monitoring solution must scale with the number of checks performed.

Monitoring Methods

There are many methods for monitoring the components that compose the enterprise. For example:

- An agent located on the component, collecting the requested information, and passing accumulated data to a central point



- A remote agent collecting data from multiple components, accumulating the collected parameters, and passing them to a central point
- A central collector which collects data from all components and processes them

There are different optimal data collection methods for different components. An SLA monitoring platform should support all methods.

Central Monitoring Data Repository

No matter which monitoring method is used, it is recommended that there is a central repository holding all the monitoring data. This repository can be used for:

- History data collection
- Trends analysis and pattern discovery
- A central location for data used for events correlations and compound conditions and more.

Monitoring Data Format

The monitored parameters data can be kept in many formats, however, there are two recommended methods for holding the data:

- Summary data should be held in a relational database, so it can easily be queried and reported.
- History data should be held in an RRD format (Round Robin Database). This method is recommended because it limits the amount of storage needed for keeping the data.

Collecting monitoring parameters and keeping them in a raw data format allows maximum flexibility, but it also becomes a liability, especially in large organizations, where the size of the data collected can increase to huge amounts. There is hardly any critical need to know what was the exact response time for a given transaction a few months ago.

Reporting Tools

SLA products should be able to display and priorities the handling of problems based on the SLA with the customers. This mechanism is not trivial, as sometimes a lesser (Silver) customer, who is near the end of his permitted handling time span, will have higher priority than a large (Platinum) customer, who just got a problem. The product should be familiar with the concept of SLA, as defined by the customer, and should act accordingly.



It is also recommended that the product has topological views that show problems based on the SLA. For example, a view that shows a division of the various SLA groups (Platinum, Gold, Silver,...) where each group is divided into areas, and each area contains the companies belonging to it.

SLA Monitoring Platform Genie

The *SLA Monitoring Platform Genie* from Jilroy Software addresses the issue of online proactive SLA monitoring by enabling real-time monitoring of large numbers of parameters, in large organizations, using a single central collection station or a relatively small number of agents to collect data from a large number of components in the enterprise. The product addresses all the existing monitoring needs of enterprises today, focusing on scale.

Addressing the Monitoring Needs

Discovering the SLA Monitored Elements

The *SLA Monitoring Platform Genie* includes a very powerful discovery tool that can extract data from almost any source (from network devices to databases and files). The product can discover the full network of the enterprise, or discover selected nodes from the whole network.

Defining SLA Parameters

The *SLA Monitoring Platform Genie* enables the user to define, for each monitored element, what its SLA parameters are. The product uses this information to determine the recommended priority of a failing node.

Increasing Number of Monitored Parameters

The *SLA Monitoring Platform Genie* enables the user to determine which parameters will be monitored, on which components, and at what frequency. The selection is very robust and very simple. The user can define which parameters will be collected from which components, using filters for discovering information from the components. The product will, based on the accepted result, know the status of the element and determine the priority required for handling an error event.

Multi-Protocol Parameter Collection

The *SLA Monitoring Platform Genie* is built to support multiple collection protocols. It currently supports the following protocols:

- ICMP
- SNMP
- SQL



- HTTP
- CSV
- Telnet
- SSH
- WMI
- TCP Ports monitoring

and others. Its support of the monitoring protocols integrates with all its other capabilities, like supporting increasing numbers of monitored parameters, ease of installation, and scale.

Simple Installation and Operation

The *SLA Monitoring Platform Genie* is built as a centralized monitoring product. It is centrally managed and controlled. It can operate on a single machine monitoring large numbers of parameters, on large numbers of components, at a high frequency. When scale is required, it supports distribution of its components to additional machines. Its installation and use 'out of the box' are very simple and enables the advanced user to tailor it to his or her exact needs.

Complex Correlations and Conditions on the Monitored Parameters

The *SLA Monitoring Platform Genie* has a built-in mechanism for defining correlation rules and conditions on the monitored data, including the historical data, so that events can be scheduled and event handlers launched, based on these correlation rules and conditions.

Flexibility

The *SLA Monitoring Platform Genie* is designed for flexibility. There are several levels where this becomes obvious:

Monitoring Parameters Selection

With the *SLA Monitoring Platform Genie*, it is easy to define which parameters will be monitored on which nodes, in a simple and robust way, that uses filters and information discovered on the monitored components. There is no need to specify which parameters will be collected from each specific node. It is possible, and recommended, to use generic rules to define these values.

Graphical User Interface Flexibility

The product is designed with SLA in mind. It contains reports that understand the SLA concept, and has a mechanism that determines the priority that should be given to failing components. The product is also supported by topological maps with color propagation showing SLA related information. The product has simple built-in tools that enable the user to tailor his User Interface to his exact needs. The user can



change the product's menu and add or remove query reports and topological maps, all of which does not require any programming skills, and can be done at site level.

The product also contains a security system that controls what each user can see. This allows customers to view SLA reports for their components only.

There is a Web based GUI designed especially for customers to see SLA reports for their monitored comments.

Product's Components Location

In order to support scale, the product is designed to enable flexibility where the product's components are located. These components can be moved from the server's central point to near by or remote computers/sites.

There is an option of locating monitoring agents in different organizations and supporting any connection mode (connection from the agent to the data collector or vice versa, according to the organizations' security policy).

Scalability

The *SLA Monitoring Platform Genie* is designed for scalability, in all the aspects mentioned above.

Component Count Scale

The product, 'out of the box', enables the monitoring components to collect data from a large number of components, from a single machine, simultaneously. However, the product supports having multiple collection processes, located either on the same machine or distributed throughout the enterprise. This capability enables scale, as monitoring different components can be distributed over the different collection processes.

The product enables collecting data from one component without effecting the collection of data from other components.

Collected Parameters Count Scale

Each collection process can collect multiple parameters from any given number of components. As described before, the scale in the number of monitored parameters is handled by adding more collection processes.

Monitoring Frequency



Since the product is designed to monitor any given nodes simultaneously, there is no theoretical limit to the monitoring frequency, and the only actual limits are CPU, memory, and bandwidth. These limits are addressed by the capability of adding more collection processes, and distributing the monitored components between them.

Correlation Rules

The correlation rules analysis is performed by a special process. This process can also be multiplied and distributed if scale is required.